

# An ADMM Algorithm for Clustering Partially Observed Networks

**Necdet Serhat Aybat**

Industrial Engineering  
Penn State University

2015 SIAM International Conference on Data Mining  
Vancouver, Canada

# Problem setting

**Goal:** Detect “communities” from partially observed “relation” graph

**Details:**

- $\mathcal{N} = \{1, \dots, n\}$  nodes of an **undirected** graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
- $(i, j) = (j, i) \in \mathcal{E}$  if nodes  $i$  and  $j$  are “related”
- $\mathcal{N}_\ell^*$ : nodes in community- $\ell$ ,  $\{\mathcal{N}_\ell^*\}_{\ell=1}^r$  **partition** of  $\mathcal{N}$ 
  - $\bigcup_{\ell=1}^r \mathcal{N}_\ell^* = \mathcal{N}$
  - $\mathcal{N}_{\ell_1}^* \cap \mathcal{N}_{\ell_2}^* = \emptyset$  if  $\ell_1 \neq \ell_2$
- $\exists p, q \in [0, 1]$  s.t.  $0 \leq q \ll p \leq 1$ ,
$$\mathbb{P}\left((i, j) \in \mathcal{E}\right) = \begin{cases} p, & \text{if } \exists \ell \text{ s.t. } i, j \in \mathcal{N}_\ell^*; \\ q, & \text{o.w.} \end{cases}$$
- $(i, j) \in \mathcal{E}$  or  $(i, j) \notin \mathcal{E}$  is **observable** with probability  $p_0$

# A popular quality function: **Modularity**

Measures the **density of edges within clusters** relative to a *random graph* with the *same node degrees*

- $\{\mathcal{N}_\ell\}_{\ell=1}^r$  a **partition** of nodes in  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
- **modularity**  $(\{\mathcal{N}_\ell\}_{\ell=1}^r) := \frac{1}{2|\mathcal{E}|} \sum_{i,j=1}^n \left( D_{ij} - \frac{d(i)d(j)}{2|\mathcal{E}|} \right) \delta(i, j; \{\mathcal{N}_\ell\}_{\ell=1}^r)$
- maximizing modularity is **NP-hard**
- **Louvain method** is a popular **greedy** alg. outperforming many others in time and modularity

# A popular quality function: **Modularity**

Measures the **density of edges within clusters** relative to a *random graph* with the *same node degrees*

- $\{\mathcal{N}_\ell\}_{\ell=1}^r$  a **partition** of nodes in  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
- **modularity**  $(\{\mathcal{N}_\ell\}_{\ell=1}^r) := \frac{1}{2|\mathcal{E}|} \sum_{i,j=1}^n \left( D_{ij} - \frac{d(i)d(j)}{2|\mathcal{E}|} \right) \delta(i, j; \{\mathcal{N}_\ell\}_{\ell=1}^r)$
- maximizing modularity is **NP-hard**
- **Louvain method** is a popular **greedy** alg. outperforming many others in time and modularity

## **Shortcoming: Resolution limit**

- **Implicit assumption**: there is a chance that any two node can be connected in the random model (**not realistic for large networks**)
- **Cliques** connected by a **single edge** would be merged as  $|\mathcal{E}| \rightarrow \infty$
- Leads to **undetactable** small communities



# Preliminary: Robust PCA

$\mathbb{R}^{n \times n} \ni D = \bar{L} + \bar{S}$  such that  $\text{rank}(\bar{L}) = r \ll n$ ,  $\|\bar{S}\|_0 = s \ll n^2$

- $\bar{L}$ : **Low-rank**, incoherent  $\Rightarrow$  **not** sparse
- $\bar{S}$ : **Sparse**, nonzero loc.  $\sim$  uniformly at random  $\Rightarrow$  **not** low-rank
- $\Omega$ : **Observable** loc.  $\sim$  uniformly at random s.t.  $|\Omega| = p_0 n^2$

Candès et al.'09 proposed solving

$$(L_\rho^*, S_\rho^*) \in \operatorname{argmin} \{ \|L\|_* + \rho \|S\|_1 : \pi_\Omega(L + S) = \pi_\Omega(D) \}$$

For  $\rho = \frac{1}{\sqrt{p_0 n}}$ ,  $\exists c > 0$  such that  $(L_\rho^*, S_\rho^*) = (\bar{L}, \bar{S})$  (**exact recovery**) w.p. at least  $1 - cn^{-10}$

**Notation:**

- $\|L\|_* := \sum_{i=1}^{\text{rank}(L)} \sigma_i(L)$ , and  $\|S\|_1 := \sum_{i,j} |S_{ij}|$
- $\pi_\Omega : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$  s.t.  $(\pi_\Omega(D))_{ij} = \begin{cases} D_{ij}, & \text{if } (i, j) \in \Omega; \\ 0, & \text{o.w.} \end{cases}$

# Robust PCA for clustering

- $D \in \mathbb{S}_n$ : adjacency matrix for  $\mathcal{G}$  such that  $\text{diag}(D) = \mathbf{1}$

$$D_{ij} = \begin{cases} 1, & \text{if } (i, j) \in \mathcal{E} \text{ or } i = j; \\ 0, & \text{o.w.} \end{cases}$$

- $\bar{S}_{ij} := \begin{cases} -1, & \text{if } (i, j) \notin \mathcal{E} \text{ and } \exists \ell \text{ s.t. } i, j \in \mathcal{N}_\ell^*; \\ 1, & \text{if } (i, j) \in \mathcal{E} \text{ and } \nexists \ell \text{ s.t. } i, j \in \mathcal{N}_\ell^*; \\ 0, & \text{o.w.} \end{cases}$  and  $\bar{L} := D - \bar{S}$ .

$$\mathcal{N}_1^* = \{1, 2, 3\}, \mathcal{N}_2^* = \{4, 5\}, \mathcal{E} = \{(1, 3), (2, 3), (2, 4), (3, 5), (4, 5)\}$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}}_D = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}}_{\bar{L}} + \underbrace{\begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}}_{\bar{S}}$$

- $D = \text{Low-Rank} + \text{Sparse}$

# Robust PCA for clustering

$$\mathcal{N}_1^* = \{1, 2, 3\}, \mathcal{N}_2^* = \{4, 5\}, \mathcal{E} = \{(1, 3), (2, 3), (2, 4), (3, 5), (4, 5)\}$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}}_D = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}}_{\bar{L}} + \underbrace{\begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}}_{\bar{S}}$$

$D = \text{Low-Rank} + \text{Sparse}$

- $\text{rank}(\bar{L}) = r$ , the number of **clusters, communities**
- $\lambda_\ell = |\mathcal{N}_\ell^*|$  for  $\ell = 1, \dots, r$ , e.g.,  $\lambda_1 = 3$ ,  $\lambda_2 = 2$
- $\bar{S}$  is **sparse** with high probability



## Related work

Chen et al.'14 showed the statistical guarantees, and proposed **RPCA-B**.

**Thm:** Let  $\gamma = \max\{1 - p, q\}$ ,  $\rho = \frac{1}{32\sqrt{p_0 n}}$ , and  $K_{\min} := \min_{\ell} |\mathcal{N}_{\ell}^*|$ .  
 $\exists C, c > 0$  such that  $(L_{\rho}^*, S_{\rho}^*) = (\bar{L}, \bar{S})$  w.p.  $1 - cn^{-10}$  provided that  
 $n \log^2(n) \leq CK_{\min}^2 p_0 (1 - 2\gamma)^2$ .

### Algorithm **RPCA-B** ( $\rho$ )

```
1: STOP  $\leftarrow$  false
2: while STOP == false do
3:    $(L_{\rho}^*, S_{\rho}^*) \leftarrow \operatorname{argmin}\{\|L\|_* + \rho\|S\|_1 : \pi_{\Omega}(L + S) = \pi_{\Omega}(D)\}$ 
4:   if  $\operatorname{Tr}(L_{\rho}^*) > n(1 + \operatorname{tol})$  then
5:      $\rho \leftarrow \rho/2$ 
6:   else if  $\operatorname{Tr}(L_{\rho}^*) < n(1 - \operatorname{tol})$  then
7:      $\rho \leftarrow 2\rho$ 
8:   else
9:     STOP  $\leftarrow$  true
10:  end if
11: end while
```



## RCPA-C model by Aybat et al.

(RPCA-C): Robust PCA for Clustering (a tighter formulation)

$$\begin{aligned} (L^*, S^*) \in \operatorname{argmin}_{L, S \in \mathbb{S}_n} \operatorname{Tr}(L) + \rho \|S\|_1 \\ \text{s.t. } \pi_\Omega(L + S) = \pi_\Omega(D), \\ \operatorname{diag}(S) = \mathbf{0}, |S_{ij}| \leq 1 \quad \forall i \neq j, \\ L \succeq \mathbf{0}_n, L \geq \mathbf{0}_n. \end{aligned}$$

Let  $\chi \subset \mathbb{S}_n \times \mathbb{S}_n$  denote the feasible region.

- Clearly,  $(\bar{L}, \bar{S}) \in \chi$
- $\chi \subset \{(L, S) : \pi_\Omega(L + S) = \pi_\Omega(D)\} \Rightarrow$  same results in Chen et al.'14

$$(L^*, S^*) = (\bar{L}, \bar{S}) \text{ w.p. at least } 1 - cn^{-10}$$

## ADMIP-C: ADMM for RPCA-C

Equivalent problem using partial variable splitting:

$$\begin{aligned} \min_{L, X, S \in \mathbb{S}_n} \quad & \text{Tr}(L) + \rho \|S\|_1 \\ \text{s.t.} \quad & X = L, \quad L \succeq \mathbf{0}_n, \quad (X, S) \in \phi. \end{aligned}$$

for  $\rho > 0$  and  $\phi$  defined as

$$\phi := \left\{ (X, S) \in \mathbb{S}_n \times \mathbb{S}_n : \begin{array}{l} \pi_\Omega(X + S) = \pi_\Omega(D), \\ X \geq \mathbf{0}_n, \quad \text{diag}(S) = \mathbf{0}, \\ |S_{ij}| \leq 1 \quad \forall i \neq j \end{array} \right\}.$$

**Objective:** Develop an ADMM with increasing penalty

## ADMIP-C: ADMM for RPCA-C

Augmented Lagrangian:

$$\mathcal{L}_\mu(L, X, S; Y) := \text{Tr}(L) + \rho\|S\|_1 + \langle Y, X - L \rangle + \frac{\mu}{2}\|X - L\|_F^2$$

- **Hard** to compute **Method of Multipliers** iterate sequence:

$$(L_{k+1}, X_{k+1}, S_{k+1}) \in \text{argmin}\{\mathcal{L}_\mu(L, X, S; Y_k) : L \succeq \mathbf{0}_n, (X, S) \in \phi\}$$

$$Y_{k+1} = Y_k + \mu(X_{k+1} - L_{k+1}), \text{ where } \mu > 0$$

# ADMIP-C: ADMM for RPCA-C

**Augmented Lagrangian:**

$$\mathcal{L}_\mu(L, X, S; Y) := \text{Tr}(L) + \rho\|S\|_1 + \langle Y, X - L \rangle + \frac{\mu}{2}\|X - L\|_F^2$$

- **Hard** to compute **Method of Multipliers** iterate sequence:

$$(L_{k+1}, X_{k+1}, S_{k+1}) \in \text{argmin}\{\mathcal{L}_\mu(L, X, S; Y_k) : L \succeq \mathbf{0}_n, (X, S) \in \phi\}$$
$$Y_{k+1} = Y_k + \mu(X_{k+1} - L_{k+1}), \text{ where } \mu > 0$$

- **Easy** to compute **ADMIP-C** iterate sequence:

$$(X_{k+1}, S_{k+1}) = \text{argmin}\{\mathcal{L}_{\mu_k}(L_k, X, S; Y_k) : (X, S) \in \phi\}$$
$$L_{k+1} = \text{argmin}\{\mathcal{L}_{\mu_k}(L, X_{k+1}, S_{k+1}; Y_k) : L \succeq \mathbf{0}_n\}$$
$$Y_{k+1} = Y_k + \mu_k(X_{k+1} - L_{k+1})$$

**Note:** ADMM is **sensitive** to  $\mu$  if  $\mu_k = \mu$  for all  $k$

- Choose  $\{\mu_k\}$  such that  $\mu_{k+1} \geq \mu_k$ , and  $\sup_k \mu_k < \infty$
- No need to search optimal  $\mu^*$



## ADMIP-C: L-subproblem

**Augmented Lagrangian:**

$$\mathcal{L}_\mu(L, X, S; Y) := \text{Tr}(L) + \rho \|S\|_1 + \langle Y, X - L \rangle + \frac{\mu}{2} \|X - L\|_F^2$$

$$\begin{aligned} L_{k+1} &= \text{argmin}\{\mathcal{L}_{\mu_k}(L, X_{k+1}, S_{k+1}; Y_k) : L \succeq \mathbf{0}_n\}, \\ &= W \text{diag}\left(\max\left\{\lambda_k - \frac{1}{\mu_k} \mathbf{1}, \mathbf{0}\right\}\right) W^T, \end{aligned}$$

$W \text{diag}(\lambda_k) W^T$  is the eigenvalue decomp. of  $Q_k^X := X_{k+1} + \frac{1}{\mu_k} Y_k$ .

**Note:**

- **No need** to compute eigenvalues  $< \mu_k^{-1}$
- Use PROPACK to compute **partial** SVD of  $Q_k^X$
- $X_k \rightarrow \bar{L}$  and  $\{Y_k\}$  bounded  $\Rightarrow \{Q_k^X\}$  **numerically low-rank** eventually
- Cheap partial SVDs in the **transient phase** for small  $\mu_k$

## ADMIP-C: $(X, S)$ -subproblem

**Augmented Lagrangian:**

$$\mathcal{L}_\mu(L, X, S; Y) := \text{Tr}(L) + \rho\|S\|_1 + \langle Y, X - L \rangle + \frac{\mu}{2}\|X - L\|_F^2$$

$$(X_{k+1}, S_{k+1}) = \text{argmin}\{\mathcal{L}_{\mu_k}(L_k, X, S; Y_k) : (X, S) \in \phi\},$$

**Closed form solution:**

$$C_k = \text{sgn}\left(\pi_{\bar{\Omega}}(D - Q_k^L)\right) \odot \max\{|\pi_{\bar{\Omega}}(D - Q_k^L)| - \rho\mu_k^{-1}\mathbf{E}_n, \mathbf{0}_n\},$$

$$S_{k+1} = \min\{\pi_{\bar{\Omega}}(D), \max\{-\mathbf{E}_n, C_k\}\},$$

$$X_{k+1} = \pi_{\Omega}(D - S_{k+1}) + \max\{\pi_{\Omega^c}(Q_k^L), \mathbf{0}_n\},$$

where  $\Omega := \bar{\Omega} \cup \mathcal{D}$  for  $\bar{\Omega} \subset \mathcal{N} \times \mathcal{N} \setminus \mathcal{D}$ , and  $Q_k^L := L_k - \frac{1}{\mu_k}Y_k$

$$\phi := \left\{ (X, S) \in \mathbb{S}_n \times \mathbb{S}_n : \begin{array}{l} \pi_{\Omega}(X + S) = \pi_{\Omega}(D), \\ X \geq \mathbf{0}_n, \text{diag}(S) = \mathbf{0}, \\ |S_{ij}| \leq 1 \quad \forall i \neq j \end{array} \right\}.$$

# ADMIP-C: Convergence

**Algorithm ADMIP-C**  $(\rho, Y_0, \{\mu_k\})$

- 1:  $k \leftarrow 0, L_0 \leftarrow \mathbf{0}_n$
- 2: **while** not converged **do**
- 3:  $(X_{k+1}, S_{k+1}) \leftarrow \operatorname{argmin}\{\mathcal{L}_{\mu_k}(L_k, X, S; Y_k) : (X, S) \in \phi\}$
- 4:  $L_{k+1} \leftarrow \operatorname{argmin}\{\mathcal{L}_{\mu_k}(L, X_{k+1}, S_{k+1}; Y_k) : L \succeq \mathbf{0}_n\}$
- 5:  $Y_{k+1} \leftarrow Y_k + \mu_k(X_{k+1} - L_{k+1})$
- 6: **end while**

$Z_k = (L_k, X_k, S_k, Y_k)$  primal-dual ADMIP-C iterates

$\mathcal{Z}^*$  set of primal-dual optimal pairs to RPCA-C:

$$\min_{L, X, S \in \mathbb{S}_n} \operatorname{Tr}(L) + \rho \|S\|_1 \text{ s.t. } X = L, L \succeq \mathbf{0}_n, (X, S) \in \phi.$$

Then  $\min\{\|Z - Z_k\|_F : Z \in \mathcal{Z}^*\} \rightarrow 0$ .



# $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ - Random Network Generation

$\mathcal{G}$ : undirected network with

- $n := |\mathcal{N}|$  nodes, and  $r$  non-overlapping communities
- $\alpha \in (0, 1]$  parameter controlling cluster sizes  $|\mathcal{N}_\ell^*|$  for  $\ell = 1, \dots, r$
- $|\mathcal{N}_\ell^*| := n_\ell = \left\lceil \frac{1-\alpha}{1-\alpha^r} \alpha^{\ell-1} n \right\rceil$  for  $\ell = 1, \dots, r$
- $\mathcal{N}_\ell^* = \left\{ \sum_{i=1}^{\ell-1} n_i + 1, \dots, \sum_{i=1}^{\ell} n_i \right\}$

**Example:**  $|\mathcal{N}| = n = 100$ , and  $r = 5$

$\alpha$	$n_1$	$n_2$	$n_3$	$n_4$	$n_5$
<b>1</b>	<b>20</b>	20	20	20	<b>20</b>
<b>0.9</b>	<b>24</b>	22	20	18	<b>16</b>
<b>0.8</b>	<b>30</b>	24	19	15	<b>12</b>
<b>0.7</b>	<b>36</b>	25	18	12	<b>9</b>
<b>0.6</b>	<b>43</b>	26	16	9	<b>6</b>
<b>0.5</b>	<b>52</b>	26	13	6	<b>3</b>



# $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ - Random Network Generation

After  $\{\mathcal{N}_\ell^*\}_{\ell=1}^r$  is generated with  $r = \lceil 0.05 |\mathcal{N}| \rceil$

## Edge generation:

- $\mathcal{E}^U \subset \mathcal{U} := \{(i, j) \in \mathcal{N} \times \mathcal{N} : i < j\}$  chosen **uniformly at random** among all subsets with cardinality  $|\mathcal{E}^U| = \lceil 0.05 |\mathcal{U}| \rceil$
- $\mathcal{E}^L := \{(i, j) \in \mathcal{N} \times \mathcal{N} : (j, i) \in \mathcal{E}^U\}$
- $\bar{\mathcal{E}} := \{(i, j) \in \mathcal{N} \times \mathcal{N} : \exists \ell \text{ s.t. } i, j \in \mathcal{N}_\ell^*\}$
- $\mathcal{E} := \bar{\mathcal{E}} \Delta (\mathcal{E}^U \cup \mathcal{E}^L)$  - symmetric difference

## Observable indices:

- $\Omega^U \subset \mathcal{U}$  chosen **uniformly at random** among all subsets with cardinality  $|\Omega^U| = \lceil p_0 |\mathcal{U}| \rceil$
- $\Omega^L := \{(i, j) \in \mathcal{N} \times \mathcal{N} : (j, i) \in \Omega^U\}$
- $\bar{\Omega} := \Omega^U \cup \Omega^L$  and  $\Omega := \bar{\Omega} \cup \mathcal{D}$

## Numerical tests: ADMIP-C vs RPCA-B

- ADMIP-C solves RPCA-C formulation (tighter than RPCA)
- RPCA-B solves RPCA with bisection on  $\rho$

Given  $L^*$  be low-rank component output, define for  $1 \leq l_1, l_2 \leq r$

$$B_{l_1, l_2}^* := \left( L_{ij}^* \right)_{i \in \mathcal{N}_{l_1}^*, j \in \mathcal{N}_{l_2}^*}, \quad \bar{B}_{l_1, l_2} := \begin{cases} \mathbf{E}_{n_\ell}, & \text{if } l_1 = l_2 = \ell; \\ \mathbf{0}_{n_{l_1} \times n_{l_2}}, & \text{o.w.} \end{cases}$$

Define  $R_{l_1, l_2} := \|B_{l_1, l_2}^* - \bar{B}_{l_1, l_2}\|_F$  for  $1 \leq l_1, l_2 \leq r$ .

The quality of clustering is measured by 3 statistics:

- $[0, 1] \ni s_{\text{in}} := \text{average of } \left\{ \frac{R_{\ell, \ell}}{n_\ell} \right\}_{\ell=1}^r$
- $[0, 1] \ni s_{\text{out}} := \frac{1}{\sqrt{\sum_{l_1 \neq l_2} n_{l_1} n_{l_2}}} \sqrt{\sum_{l_1 \neq l_2} R_{l_1, l_2}^2}$
- $[0, 1] \ni s_{\text{f}}$ : fraction of correctly recovered clusters

# Numerical tests: **ADMIP-C** vs **RPCA-B**

10 random  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  for each  $(n, \alpha)$ , and  $p_0 = 0.9$

$n$	$\alpha$	ADMIP-C			RPCA-B			RPCA		
		$s_{in}$	$s_{out}$	$s_f$	$s_{in}$	$s_{out}$	$s_f$	$s_{in}$	$s_{out}$	$s_f$
100	1	0	0	100	1	0	100	1	0	100
	0.9	0	0	100	1	0	100	3	0	100
	0.8	0	0	100	0	0	100	16	0	82
200	1	0	0	100	0	0	100	18	0	94
	0.9	0	0	100	0	0	100	48	0	51
	0.8	3	0	99	21	19	0	53	0	45
300	1	0	0	100	0	0	100	93	0	0
	0.9	3	0	100	21	19	0	65	0	33
	0.8	8	0	97	22	19	0	67	0	33
400	1	1	0	100	0	0	100	100	0	0
	0.9	6	0	98	23	19	0	71	0	30
	0.8	10	1	94	20	20	0	71	0	30
500	1	2	0	100	1	0	100	100	0	0
	0.9	5	0	98	24	19	0	74	0	26
	0.8	10	0	92	24	19	0	74	0	25

Table: Mean values for  $s_{in}$ ,  $s_{out}$ , and  $s_f$  in %, and  $\rho = \frac{1}{\sqrt{n}}$



# Numerical tests: ADMIP-C vs RPCA-B

5 random  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  for  $n \in \{500, 1000\}$ ,  $\alpha = 0.95$ , and  $p_0 = 0.9$

$n$	#	ADMIP-C					RPCA-B					
		$s_{in}$	$s_{out}$	$s_f$	cpu	svd	$s_{in}$	$s_{out}$	$s_f$	cpu	svd	$it_B$
500	1	4	0	100	4	30	22	18	0	14	61	3
	2	3	0	100	4	31	21	18	0	13	58	3
	3	2	0	100	4	33	20	18	0	13	58	3
	4	2	0	100	3	29	21	18	0	13	60	3
	5	3	0	100	3	31	22	18	0	14	60	3
1000	1	5	1	98	27	40	27	22	0	159	92	4
	2	3	0	100	26	37	26	22	0	163	92	4
	3	4	0	100	26	39	27	22	0	162	92	4
	4	4	1	100	27	41	26	22	0	161	90	4
	5	4	0	98	26	38	26	22	0	163	93	4

Table: Recovery and runtime statistics:  $s_{in}$ ,  $s_{out}$ ,  $s_f$  in %, cpu in sec.



# Numerical tests: ADMIP-C vs RPCA-B

---

## Summary:

- Increasing  $n$ , decreasing  $\alpha$  adversely affects all methods
- Failure caused by  $s_{\text{out}}$  for RPCA-B, by  $s_{\text{in}}$  for RPCA and ADMIP-C
- RPCA cannot detect small clusters due to high  $s_{\text{in}}$
- RPCA-B reduces  $s_{\text{in}}$  at the cost of  $s_{\text{out}}$ : merging small clusters
- ADMIP-C correctly identifies 92% all the time

## Numerical tests: ADMIP-C vs Louvain

Measures for comparing similarity of  $\mathcal{C} = \{\mathcal{N}'_i\}_{i=1}^{r'}$  and  $\mathcal{C}^* = \{\mathcal{N}^*_j\}_{j=1}^r$ :

$n'_i := |\mathcal{N}'_i|$ ,  $n_j := |\mathcal{N}^*_j|$  s.t.  $n = \sum_{i=1}^{r'} n'_i = \sum_{j=1}^r n_j$ , and  $m_{ij} = |\mathcal{N}'_i \cap \mathcal{N}^*_j|$

- Jaccard Index:

$$\text{JI} := \frac{\# \text{ node pairs in same clusters both in } \mathcal{C} \text{ and } \mathcal{C}^*}{\# \text{ node pairs in same clusters either in } \mathcal{C} \text{ or } \mathcal{C}^*}$$

- Normalized Mutual Information:

$$\text{NMI} := \frac{\mathcal{I}(\mathcal{C}, \mathcal{C}^*)}{\sqrt{\mathcal{H}(\mathcal{C})\mathcal{H}(\mathcal{C}^*)}}, \text{ where}$$

- Mutual Information:  $\mathcal{I}(\mathcal{C}, \mathcal{C}^*) = \sum_{i=1}^{r'} \sum_{j=1}^r \frac{m_{ij}}{n} \log_2 \left( \frac{m_{ij}/n}{n'_i n_j / n^2} \right)$
- Entropy:  $\mathcal{H}(\mathcal{C}) = - \sum_{i=1}^{r'} \frac{n'_i}{n} \log_2 \left( \frac{n'_i}{n} \right)$ ,  $\mathcal{H}(\mathcal{C}^*)$  defined similarly.

- PERC := proportion of exactly recovered clusters

# Numerical tests: **ADMIP-C** vs **Louvain**

---

## Experimental setup:

- $n \in \{100, 200, 300, 400, 500\}$  and  $\alpha = \{1, 0.9, 0.8\}$
- 20 random  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  generated for each  $(n, \alpha)$
- Output of **Louvain** depends on the ordering of nodes
- For each  $\mathcal{G}$ , 200 **node orderings** generated uniformly at random



# Numerical tests: **ADMIP-C** vs **Louvain**

		Louvain					ADMIP-C				
		100	200	300	400	500	100	200	300	400	500
	$n \backslash \alpha$										
	JI	1	99.9	99.1	96.5	92.8	85.0	100	100	100	100
0.9		99.9	95.0	78.4	67.4	59.7	100	100	99.9	99.9	99.7
0.8		99.7	85.0	73.6	68.7	67.5	100	99.9	99.9	99.9	99.9
NMI	1	99.9	99.8	99.5	99.0	98.1	100	100	100	100	99.9
	0.9	99.9	98.8	94.4	91.1	88.1	100	100	99.9	99.9	99.6
	0.8	99.9	93.9	88.6	86.5	85.8	100	99.9	99.8	99.6	99.6
PERC	1	99.9	99.0	96.2	92.2	83.3	100	100	100	100	99.4
	0.9	99.9	91.1	51.8	30.9	17.2	100	100	99.3	97.5	90.2
	0.8	99.5	50.1	17.1	10.3	5.73	100	98	93.3	81.8	84.2

Table: The mean values for 3 measures in % when  $p_0 = 0.9$ .

- Increasing  $n$ , decreasing  $\alpha$  adversely affects all methods
- **Louvain**: small clusters swallowed by large ones in both cases
- **ADMIP-C**: detects small clusters even for **small**  $\alpha$
- **ADMIP-C**: creates isolated nodes for **large**  $n$